

Spacecraft Testing Programs: Adding Value to the Systems Engineering Process

Keith J. Britton
NASA, Kennedy Space Center

Dawn M. Schaible
NASA, Langley Research Center

ABSTRACT

Testing has long been recognized as a critical component of spacecraft development activities — yet many major systems failures may have been prevented with more rigorous testing programs. The question is why is more testing not being conducted? Given unlimited resources, more testing would likely be included in a spacecraft development program. Striking the right balance between too much testing and not enough has been a long-term challenge for many industries. The objective of this paper is to discuss some of the barriers, enablers, and best practices for developing and sustaining a strong test program and testing team. This paper will also explore the testing decision factors used by managers; the varying attitudes toward testing; methods to develop strong test engineers; and the influence of behavior, culture and processes on testing programs.

KEY WORDS: Risk, Integration and Test, Validation, Verification, Test Program Development

INTRODUCTION

One of the roles of testing in a successful program is to allow an opportunity for system errors to be discovered and corrected before the system is put into operation. Testing is performed for risk mitigation; it serves as an insurance policy against failed missions. Yet many operational failures in different industries could potentially have been avoided with more thorough testing during development phases. If testing is recognized as such a benefit, then why is more testing not performed? Given unlimited time and unlimited budget, more testing could be performed. Since it is unfeasible for programs to perform exhaustive testing, there must be a balance between too little testing and too much testing. Not enough testing adds risk to a program, while excessive testing can be very costly and may add unnecessary run-time on the equipment. Determining exactly how much testing is just enough is an extremely difficult question for many program managers and other decision makers. The decision process often appears arbitrary and has received little attention in the past. The objective of this paper is to explore some of the barriers, enablers and best practices for developing and sustaining a strong test program and testing team.

METHODOLOGY

This paper is a result of previously completed thesis work that researched the hypothesis that many decision makers do not utilize a holistic approach in addressing testing requirements (Britton and Schaible 2003). In addition, some decision makers are not fully aware of the

influences inherent in the decision making process. To investigate these theories, expert knowledge regarding integration testing was captured through a series of interviews with recommended experts in the testing discipline. The interview process was intended to solicit the tacit knowledge of the interviewee and was not structured to obtain data for statistical analysis. Fourteen experts were interviewed, representing a cross-section of the testing community (from test engineers to test program managers, both from government agencies and contractors). This set was deemed not large enough to constitute a statistically significant sample of the overall population. The lack of statistical significance, however, does not detract from the qualitative value of the knowledge obtained through the interviews and subsequent response analysis.

Development of the common themes from the interviews was accomplished from the perspective of the researchers' testing experiences. An internal perspective was developed from the researcher with extensive experience in human-rated spacecraft testing, while an external perspective was developed from the researcher with limited testing experience. These differing approaches to the interview responses provided a deeper understanding of the knowledge captured, and resulted in a broad set of recommendations for future test programs. Finally, after the interview data analysis was completed, a collection of overall themes and insights emerged. The data analysis, along with the literature information, was used to develop the common themes and findings described in this paper.

BARRIERS TO ADEQUATE TESTING PROGRAMS:

A number of barriers to adequate testing programs emerged from the interviews with testing experts. It should be noted that these barriers can be highly inter-related or coupled. Any combination of the barriers may be involved in a specific system failure that is linked to inadequate testing. The most significant barriers are described below.

Subjectivity of test requirements development:

Typically, the actual test-requirement decision process is not documented as a formal process, but relies heavily on the judgment of the decision makers, adding to the overall subjectivity. Due to this subjectivity, testing practices are not consistent across the aerospace industry.

The essence of this subjectivity comes partly from the risk identification and mitigation processes and partly from the systems requirements processes. From a systems perspective, it is impossible to relate risk factors perfectly to testing requirements, due to the complexity of the system interactions and the difficulty in recreating the spacecraft operational environment for pre-flight ground tests. Additionally, just as the system architect must answer the question "Is this what the customer wants?", the testing organization must answer the question "How do I prove the system is operating as expected?" Because each system is different and has different system requirements, the test requirements are different as well. These differences in system and test requirements make it difficult to standardize testing programs (Britton and Schaible 2003, 27).

Finally, the inconsistent nature of testing practices across the aerospace and commercial aircraft industry contributes to the subjective nature of an individual test program. The most notable inconsistency is in the way terms are used to describe test activities. Another source of

inconsistency is the wide variation in the way projects are organized within and across programs and companies. A third inconsistency is in the process of defining, implementing and tracking test requirements. All programs treat testing differently due to unique requirements and conditions.

Paradoxical nature of testing:

Testing alone does not make a project successful, but it does raise confidence in the likelihood of success. Despite this value-added attribute, testing is often regarded as a drain on program resources, rather than a valuable undertaking.

Program success is not only based on the successful operation of the system, but also on whether it is completed on time and within budget. Testing is often considered an insurance policy against future failures. Program managers must perform a risk-benefit analysis to determine the wisest use of program resources. Even on the technical side, what constitutes a successful test is itself a paradox. A hardware test that uncovers multiple problems can be considered just as successful as a test that verifies the flawless operation of a perfectly designed and built system. This paradox can create a managerial view of testing as an unrecoverable program cost rather than a value-added proposition of long-term program cost avoidance. True costs for testing are not calculated in terms of cost prevention, only in terms of actual costs to perform the test. Hence, the cost savings of identifying an error on the ground, and fixing it before launch, are often not considerations in determining a test program's success.

Vulnerability to changes and cutbacks:

Major systems and sub-systems include testing activities at the end of their development phases, yet the budget and schedule pressures that often occur are most keenly felt at the end of development programs. The dilemma of having to allocate diminishing resources (time and money) at the end of a development cycle makes testing activities the most vulnerable to cutbacks by program managers.

One source of this vulnerability is incomplete test requirements. Very often, the actual design is in flux due to continuously changing system requirements. As budget and schedule pressures increase during the development phase, managers tend to accept more risks and are therefore willing to agree to more cutbacks in testing. It can be concluded that, the more stringent the requirements, the less vulnerable the test program will be to these reductions in testing. Again, as budget and schedule pressures are most keenly felt at the end of a development program, testing is extremely vulnerable because it is one of the only remaining opportunities left to compensate for schedule delays or cut over-runs. In order to combat this temptation, some test programs are budgeted early in a program, and the budget is placed in a reserve status. However, if proper care is not taken to preserve the funding, the test budget can be decimated by the time the testing is scheduled to begin.

Inadequate attention to testing:

Testing, in general, does not receive as much attention as it should. Testing is often overlooked in the early planning phases of a project, current literature, training programs, academia, and organizational status.

A common theme of the interviews with the testing experts revealed that as an activity as well as an organization, testing does not receive the same attention as other elements of a program. In general, testing is recognized as an important part of a project. As mentioned earlier however, because testing typically occurs at the end of development phases, it does not receive as much emphasis during the initial planning phases as it should. The lack of emphasis can be in the form of time, budget, or forethought spent on the testing during the project planning process. Understandably, design has to take precedence in defining the function and form of the system in the beginning of development. It is only after the program is well underway that testing begins to receive the consideration it deserves. If adequate consideration is not given to testing at the beginning of a program, it may be too late to ensure that test planning and implementation is both thorough and efficient. Many of the interviewees expressed frustration with this trend. One expert summed it up as “not enough emphasis is placed early enough on high-quality test requirements, equipment, and procedures. We make it work but it’s not efficient” (Britton and Schaible 2003, 29).

Lack of training and mentoring was repeatedly mentioned as a concern within the testing area. Much time and effort for training and mentoring has been placed in other areas of program and project planning, but testing has not received its fair share. According to some of the expert interviewees, the lack of a detailed and documented testing strategy has meant that each program, or new program participant, has had to relearn important lessons from the past. The experts recommended that more should be done to capture these best practices in order to provide decision makers with some guidelines for test planning.

Finally, several of the testing experts suggested that the deficiencies in test planning might be an organizational issue. Insufficient forethought is given to how a program will be structured to allow for the most effective test planning and implementation. Many of the testing experts reported that test engineers do not hold the same status or standing of other engineers within the program. These test engineers are often not as respected as other engineers are and as such, do not receive the same opportunities or receive as much attention.

Optimism varies with organizational position:

The level of optimism in the adequacy of the test programs may increase up the chain-of-command to the managers. There are several possible reasons for why an individual’s outlook on testing changes with their organizational position. The first is an individual’s ability to take a holistic view of the project. Senior managers usually have a broader view than many of the engineers at the lower organizational levels. The system and test engineers at lower organizational levels are primarily concerned with their own system and its successful operation. Managers, on the other hand, must balance the entire system, including limited resources, political factors, and a larger constituency. In turn, the program manager’s risk-aversion level is different from that of the lower-level engineers. The program managers must balance the trade-offs and, as such, are perceived to have a higher tolerance for risk. These project managers do understand, however, that they are responsible for accepting risk as part of the trade-off decisions. As one project manager stated “I was willing to take the risk of not testing what I flew. As the project manager for the ...mission, I was the one who ultimately decided what risks to take...” (Britton and Schaible 2003, 37).

Another possible reason for the differing level of optimism is that the understanding of the detailed technical information is most likely not consistent across the organization. It is reasonable to think that the lower-level engineers understand the operation of their system better than the managers. Information may not be conveyed to managers in a way that allows proper risk evaluation. When this less-than-perfect knowledge is combined with the other external pressures, managers may be accepting more risk than they realize.

The established culture also plays a role in the differences in optimism. An organization may have a culture of risk-taking, while other may be risk adverse. Depending on the culture of a particular project, acceptable risk level changes. In addition, many organizations are very accustomed to fixing technical problems as soon as they are discovered, but the process for reversing bad decisions made early in a program are not formalized. It should be noted that program and test managers generally do not make bad decisions intentionally, or out of incompetence. Early in a program, all the information may not be available or may not be accurate. Managers make the best decisions they can at the time, but these decisions are rarely re-evaluated. The lower-level engineers may recognize these as bad decisions, but the managers may not even be aware that there is an issue. The tendency is to just work around the issue, rather than change the ground rules.

Current methods of tracking testing costs are not sufficient:

Actual cost figures for spacecraft testing programs are very difficult to determine because they are not typically tracked as discrete line items in spacecraft programs. Pure test costs are often obscured by engineering and operational funding, making them susceptible to misinterpretation.

Test costs are not consistently defined from one program to another, making comparisons difficult between programs. Some have attempted to establish heuristics for estimating testing costs for use in developing cost proposals. These estimating “rules of thumb” are subject to negotiation during the proposal development phase, as well as during final contract negotiations with the customer. During the actual performance of testing activities, detailed financial information is more readily available, but again this information is not easily broken down to actual test costs. Given the proprietary nature of cost data, the sharing of the information across different organizations and companies can be difficult and hampers the ability to improve on the “rules of thumb”.

Better estimating and tracking of test activities is needed to establish a consistent estimating tool that can be used in early program planning. Accurate testing cost data can also be used to establish the savings that testing brings to a program in the form of cost avoidances from errors that may have been overlooked had the testing been eliminated. Life-cycle cost consideration is another deficient area in making test decisions. For human-rated spacecraft it is often attractive to forego a test and accept the risk that an on-orbit repair may be necessary. However, when the cost of on-orbit repairs is considered in the decision, it often becomes apparent that performing a test is money well spent in the end.

ENABLERS FOR GOOD TEST PROGRAMS

While testing practices vary, decision factors do not:

Contrary to what may be expected, there is consistency in the decision factors used by the various decision makers. These common decision factors can be divided into technical (safety, risk, and confidence building) and non-technical (resource availability, process, individual decision-making behavior, and political/cultural influences). In addition, decision makers consider various sources of uncertainty, especially the effects of reuse, an inexperienced team, and potential unexpected emergent behavior.

Three technical decision-making factors have been identified that deal directly with the spacecraft systems – safety, risk, and confidence level. Safety is never intentionally compromised in the decision-making process. This is not to say that safety in itself is never compromised. Many mission or system failures could have been avoided had more, or better, testing been performed. A major difficulty that program managers face is the understanding of what conditions constitute a threat to safety. While program managers know that safety is the first priority, they may not know when their decisions actually compromise safety.

The higher the risk level, whether in the form of likelihood or consequence, the more likely testing will be required as a form of risk mitigation. The risk assessment, and minimally acceptable risk, depends on the particular project and the circumstances. There are often pre-conceived biases of what the likelihood and consequences are for particular projects. These pre-conceived notions have led to some general misconceptions. For example, some have stated that human-rated and expendable spacecraft must be treated differently. In reality, the same factors apply to both types of spacecraft, but the major difference is in the respective consequences. The ability to detect and correct a failure of a spacecraft post launch must be included in determining the consequence level. Each project must assess the relative risk; however, the resulting actions for testing will differ based on the circumstances.

Another prime factor that may not be as obvious to the decision maker is the amount of uncertainty present. Uncertainty defines how well the risk can be determined and is based on the amount of system knowledge that is available at the time of the decision. For example, the amount of previous experience with a design or the amount of testing previously completed contributes to the overall system knowledge. As system knowledge increases, confidence in the system's operation also increases. Many other factors also contribute to the overall level of confidence in a system. The sources of uncertainty in the system's operation can include the following: design maturity, new technology, system complexity, new application of heritage hardware/software, previous testing, fidelity of data/results, and team experience. It is important to note that it is these factors that are often overlooked or underestimated in the decision-making process. By not considering these factors, many program managers have encountered difficulties. Testing is a primary means by which uncertainty can be reduced.

Upfront planning is a key to success, but be prepared for change:

Early planning in both systems engineering and testing is necessary to manage the complexity of human-rated spacecraft programs. The early involvement of the test organization is also essential

in establishing a set of mandatory test requirements that will be less vulnerable to changes later in the program.

The success of a testing program depends, in part, on the quality of the test requirements. However, early emphasis on high-quality test requirements is often missing from program planning decisions. Getting an early start on philosophical and policy decisions is necessary to establish the right priorities regarding the level of testing and criticality level of the requirements.

The early involvement of the test organization is also essential in establishing a set of firm test requirements (often referred to as a critical test list) that will be less vulnerable to changes later made by the program. By establishing critical test requirements that must be performed, program managers and test managers are in a better position to respond to program changes. Less critical requirements can also be established early in the program, but they can be satisfied by other methods of verification, or not performed in favor of cost or schedule considerations. This is appropriate as long as the risks associated with not performing the test are acceptable.

Early decisions regarding the testing approach are made at the program-management and systems-engineering levels. The program managers are responsible for overall resource allocation and cross-system integration decisions, and they must also consider recommendations from systems engineers. The systems engineers in turn base their recommendations on the inputs of the various hardware and software owners. This simplified decision hierarchy involves many organizations and groups assigned to various aspects of the program. Decision makers at all levels must consider a wide range of variables when addressing testing programs. These variables include when to perform tests, what tests to perform, where to test, fidelity of tests, the amount of software independent verification and validation to perform, and how much retest should be allocated as contingency.

Upfront planning also allows the opportunity to begin testing early. An advantage of early testing is to verify component behavior under system-level conditions. Gaining this knowledge of actual system operations and behaviors is useful in driving down the level of uncertainty in the system. This testing also allows identification of transients at an early stage. The testing experts regarded this early emphasis as necessary to begin the process of addressing the “little glitches,” or unexpected emergent behavior, that must be understood as the overall system is developed. Ignoring these small details can lead to maintenance issues during operation, or even worse, system failures.

The process of identifying unexpected emergent behaviors must begin at an early stage in the program to ensure that as many problems as possible are recognized and addressed. Analytical methods can be applied to early systems to find and correct possible interactions before the design is complete. Analysis alone will not identify all unexpected emergent behaviors, and testing is necessary in order to ensure that those interactions that do present themselves will not pose an unacceptable situation for the spacecraft or crew.

Allowing enough time in the testing schedule to address these unexpected emergent behaviors and other contingencies is also important. Because testing occurs at the end of the development phase, the remaining available schedule is tightly controlled. The objectives of later tests may

become vulnerable if too much time is spent correcting errors uncovered by the early test procedures. While schedules are typically optimistic, testing will find errors and the planning should account for these contingencies.

Finally, early planning also allows program managers to develop enough flexibility in the test program to make adjustments during the later phases. Often, windows of opportunity develop from changes in other parts of the program (i.e., delays in launch schedule) that can create time for more testing. If a prioritization of additional test activities is not readily available, these opportunities can be lost.

Testing is more art than science:

While formal training provides a solid foundation in the concepts and methodologies of developing test programs, experience and mentoring are more important in developing test-engineering expertise. To maintain the knowledge base and core capabilities, however, test engineering should be treated as a valid discipline with the same prestige level as other engineering disciplines.

In general, college curriculums and industry only give testing superficial emphasis. Testing is mentioned as a necessary component of system development, but does not include a discussion of actual test implementation. Formal training is also not provided either at the collegiate level, or in most industries.

Because testing practices are not formally documented, the most valuable knowledge is retained within the individual experts as tacit knowledge. As organizations change from experience-based knowledge to process-based knowledge, there is a danger of losing the testing expertise that resides within the organization. Once this knowledge is lost, the chances of repeating past mistakes increases.

Providing test engineers with a viable career path will discourage them from moving to other assignments and will serve to attract talented new engineers to the profession. Currently, test engineering does not receive the recognition appropriate to the amount of creativity and difficulty involved as compared to design engineering. Test engineers need to know the system design and then must be more creative in attempting to identify system faults. Finally, any organizational structure should attempt to capture and retain the core knowledge of its testing group. Mentoring and on-the-job-training are two ways of retaining testing knowledge and passing it along to new programs.

RECOMMENDATIONS

The following recommendations are offered for improving test programs:

- Establish improved training and mentoring programs for test engineers. Organizations should also evaluate the feasibility of establishing a small testing corps that remains intact and moves from one project to another, augmenting the resident testing teams of each project. This testing corps would mentor the other engineers on the project, while learning new skills and best practices as well. Test expertise would proliferate throughout the organization with

this approach. Alternatively, separate test organizations could be established within the individual projects in order to develop skills, facilitate knowledge transfer, improve communication, and establish consistent test philosophies. Establishing a separate test organization, however, requires increased integration and communication.

- Consider test engineering a valid discipline with a viable career path for test engineers. This would help to improve the current level of prestige that testing receives within the organization and would assist in retaining the critical test engineering skills.
- Include testing in the earliest stages of the spacecraft's development process. Proper system and test engineering representation in the design definition phase will increase the fidelity of the test requirements and effectiveness of test programs. In addition, having previous hardware and software experience can be very important in recognizing system trends and unexpected emergent behaviors during the later phases of the program.
- Recognize and understand personal risk-tolerance levels and how individual decision-making styles affect decisions. Furthermore, decision-making styles should be matched to specific types of management positions. A balanced team of risk-averse and risk-tolerant personalities will provide a collaborative environment that will place proper emphasis on all of the decision factors being considered.

SUMMARY

With the increased complexity of today's modern systems, testing programs will become more important than ever as a tool for improving mission success. With this increased complexity, it becomes necessary to identify inherent system behaviors at every stage of the system development cycle. Understanding and managing the barriers and enablers of good testing programs is important. Further research into system complexity and how testing can be used to discover unintended interactions, however, is warranted. Advances in technology that can improve modeling of systems at an earlier stage should be explored and the role of test-program development integrated into the process.

REFERENCES

Britton, Keith J., and Dawn M. Schaible, "Testing in NASA Human-Rated Spacecraft Programs: How much is just enough?" *Massachusetts Institute of Technology Thesis* (2003).

BIOGRAPHIES

Keith Britton

Mr. Keith Britton is a Senior Project Manager at NASA's Kennedy Space Center where he is currently assigned to the Construction of Facilities Project Planning and Integration Office. Mr. Britton manages a team of project managers in the planning and implementation of various facility and infrastructure projects in support of NASA's space exploration missions. During his twenty-year NASA career, Mr. Britton has held a wide range of technical and project management positions relating to ground support equipment development, payload operations, and construction of facilities. These assignments supported the International Space Station and Space Shuttle programs as well as various payload missions and institutional programs. Mr. Britton is a graduate of NASA's Project Management Development Program — Accelerated Leadership Option (ALO) which included a two year academic assignment at the Massachusetts Institute of Technology followed by a one year industry assignment with Lockheed Martin Aerospace Company where Mr. Britton served as the Deputy Air Vehicle Program Manager on the C-5 Modernization Program.

Dawn Schaible

Ms. Dawn Schaible is currently the Manager of the Systems Engineering Office for the NASA Engineering and Safety Center (NESC) at the Langley Research Center. Ms. Schaible began her career with NASA at the Kennedy Space Center (KSC) in 1987, where she served as a Space Shuttle Orbiter Environmental Control and Life Support Systems (ECLSS) Engineer. In this role, she led the ECLSS ground processing activities for the Orbiter Endeavour. In 1996, Ms. Schaible joined the International Space Station (ISS) Hardware Integration Office, where she served as the Lead Test Engineer for the "Unity" Node and U.S. Laboratory "Destiny" modules. In 2000, Ms. Schaible was selected to serve as Chief, Integration Branch for the ISS/Payload Processing Directorate. Ms. Schaible completed the Systems Design and Management Program at the Massachusetts Institute of Technology, where she received a M.S. degree in Engineering and Management. Ms. Schaible previously received a B.S. degree in Mechanical Engineering from Bradley University and a M.S. degree in Space Systems Operations from the Florida Institute of Technology.